Graphic Communications Computer Association, an Affiliate of Printing Industries of America, Inc.
1730 North Lynn Street/Suite 604, Arlington, Virginia 22209. Phone (703) 841-8160.

GenCode Overview and Update

By JOSEPH V. GANGEMI
Vice President, Systems and Programming
Lehigh/Rocappi, Inc.

What is generic coding?

Generic coding is a concept which incorporates a definition of the intellectual content of data into a file in machine processable form. Generic coding is a concept, not a programming language. It is not even a new concept. It is more a new phrase being used relative to text processing than an original idea. If you think about it, generic coding is a traditional computer coding scheme for non-text applications. When a programmer says positions 1 to 10 of a record contain the part number of a product, and positions 11 to 15 are the price, he is really doing generic coding — incorporating a definition of the intellectual content of the data into the file by giving a meaning to the location of the data within each record. This form of coding is considered to be implicit coding because no actual code exists in the file to impart the meaning of the data.

In the case of running text files, implicit coding does not work since there is no fixed location within a record that can be assigned a generic meaning. In running text files some sort of embedded explicit code must be used to give a meaning to the data being processed.

As computer equipment evolved from primarily number processors to character processors with the ability to handle text-like data, many types of coding systems also evolved. Each coding system was unique and was designed to control some sort of output process. While it is true that this coding also had a tendency to impart the intellectual content of the data and was therefore generic coding (remember this is not a new idea), it did it as a secondary function, not a primary one.

Most people also confuse generic coding with generalized coding like that found in computerized composition systems. Generalized coding has a different meaning.

Generalized coding is a concept which incorporates in a file codes with no predetermined function. These codes take on a meaning or function at processing time and are unique and possibly generic within a single file but are not consistent in generic meaning across a broad range of files. These codes are generalized from the standpoint of output formatting. The same code in different files has a completely different meaning both functionally and generically.

The actual concept of generic coding began to evolve in large shops with large computer systems that found themselves handling text in different ways for different device types. By applying the traditional techniques of coding data to text files, several benefits could be achieved.

First, the data was separated from the application in which it was eventually to be used, a traditional data base concept. This permitted the use of the same data as input to each of the device-dependent applications by developing programs which converted generic codes to the corresponding application-dependent codes.

Operator training was reduced because only one coding system had to be learned. The same operators could prepare input for all output devices.

The data became more valuable because the intellectual content of the data was being captured. The same data could be used in applications with widely different purposes, from format-oriented applications such as preparing computer printouts, including the use of laser printers and microfilm or fiche, to content-oriented applications such as information retrieval for data analysis and reporting and electronic publishing.

However, the traditional problem of coding systems still is occurring. Every system is different because no standards have been established for going about doing generic coding. That is when GCCA decided to get involved and develop GenCode. In an effort to head off the problem before it became too big to surmount, GCCA established a committee to develop standards for the generic coding of text documents.

GenCode is the GCCA coding system which applies the principles of generic coding to text documents. The coding system's goals are to permit the capture of text, including its intellectual content, to be device- and application-independent, and to permit the transfer of data between different systems.

GenCode is not a computer program, but a system for capturing data in a way suitable for processing by computer. GenCode does not presuppose that any program exists that can process the language, only that a program can be developed to process it.

What is GenCode? GenCode is a generalized markup language for coding text documents. It is not a processing language. A markup language identifies data; a processing language formats it. It is an open-ended language which can be

This paper was originally presented at GCCA's 1981 Data Base Publishing Conference in Washington, D.C., April 22, 1981.

expanded to meet the needs of any type of document. It is adaptable by each user to his own specific document type.

What does GenCode consist of? GenCode consists of a set of rules or a syntax for the codes used in a text file. It consists of a definition of the structure for the components of a document and a recommended set of tags for each component — a starter set for the user to begin coding.

In order to define a syntax for GenCode, the GenCode committee tracked what was being developed by the ANSI committee X3J6. This committee is developing national standards for the processing of text and a standard language for text processing. The GenCode committee became a working group of the ANSI X3J6 committee and has recommended a standard syntax for a generalized markup language for text files. It appears at this time that this syntax will be accepted as a national standard by ANSI.

The next task of the GenCode committee is to define the components and structure of a document. Text by its very nature has a hierarchical structure. A book is made up of zones: front matter, body matter, and rear matter. Each of these zones has segments. Front matter may consist of a title, a preface, and a table of contents. Body matter may consist of chapters. Rear matter may contain an appendix and an index. (See illustration 1.)

Within each segment there are many types of lower level elements such as heads, lists, paragraphs, footnotes, tables, highlighted phrases, and so on. Each of these elements has interrelationships or attributes which affect its meaning and content relative to other elements in the document. For example (Illustration 2), look at a Chapter within the body matter of a document. As you can see, this chapter has several main heads. These heads have subordinate paragraphs or lists or other heads. These lower level elements have similar elements subordinate to them. All of these interrelationships must be coded into a text file in order for the data to be properly marked-up.

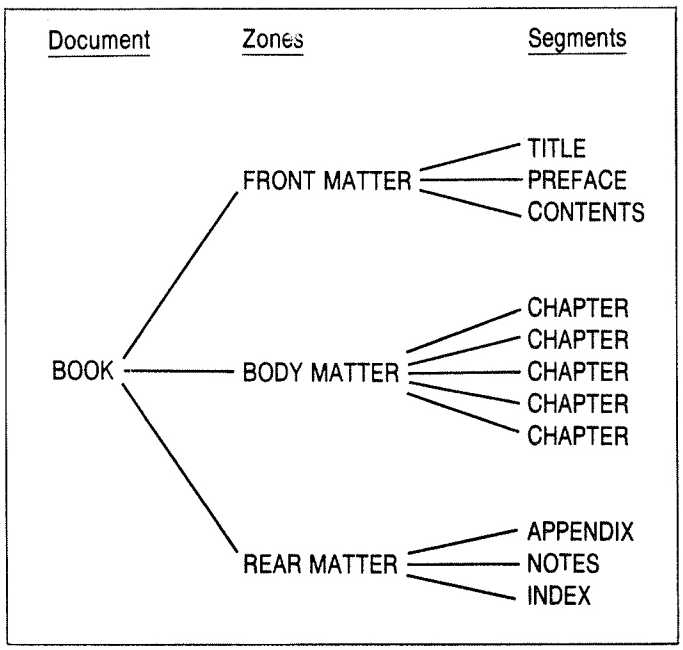

Illustration 1. A sample document hierarchy

The GenCode committee is undertaking the task of defining all of the elements in a document and their interrelationships to one another. At present the committee has defined several key concepts of data interrelationship, such as the order of elements, element levels, and recursion.

An element may be sequential or asequential. Sequential elements are those which follow the normal flow of text and appear one after the other. Asequential elements, such as footnotes, are those which need not appear in the sequential order of text at the point at which they occur in the data file.

Elements may be nestable or non-nestable. Nestable elements are those sequential elements which are embedded within some other element. An example of a nested element might be a footnote call-out or a highlighted (italicized) phrase. Non-nestable elements may not be embedded in another element. Examples would be a Title or Table of Contents or a Chapter or an Index.

Elements may have levels so that a relationship of subordination can be expressed.

Recursion is a special characteristic of text which is applied to entire documents or zones. Recursion is the nesting of a document or zone within another document or zone. Recursion permits the bringing together of separate documents into a single new document.

The GCCA is updating the Primer which explains in more detail these concepts. The Primer also includes a starter set of codes which can be used to code a file generically. The user can change or augment this starter set to meet his own document requirements.

Over the past year, the GenCode committee has made a lot of progress in its development of a generalized markup language. Here is where we stand now:

1. We have been recognized by and are a working group for the ANSI committee X3J6 which is developing national standards for text processing.
2. We have developed a syntax for a generalized markup language which has been proposed as a national standard.
3. We have defined the basic structure of a text document.
4. We have defined some of the interrelationships between document elements. We shall continue with these definitions as we proceed further with our work.
5. We are in the process of defining standard document components and elements relative to their generic meaning within a document.

Why are we doing all of this? We see an increase in the volume of text being captured for computer processing of one sort or another every day — from computer typesetting to electronic publishing, from large mainframes to personalized minis. We see more diversified use of the same text in order to increase its value, to offset the cost of its capture. As the cost of computers continues to decrease the marriage between tele-

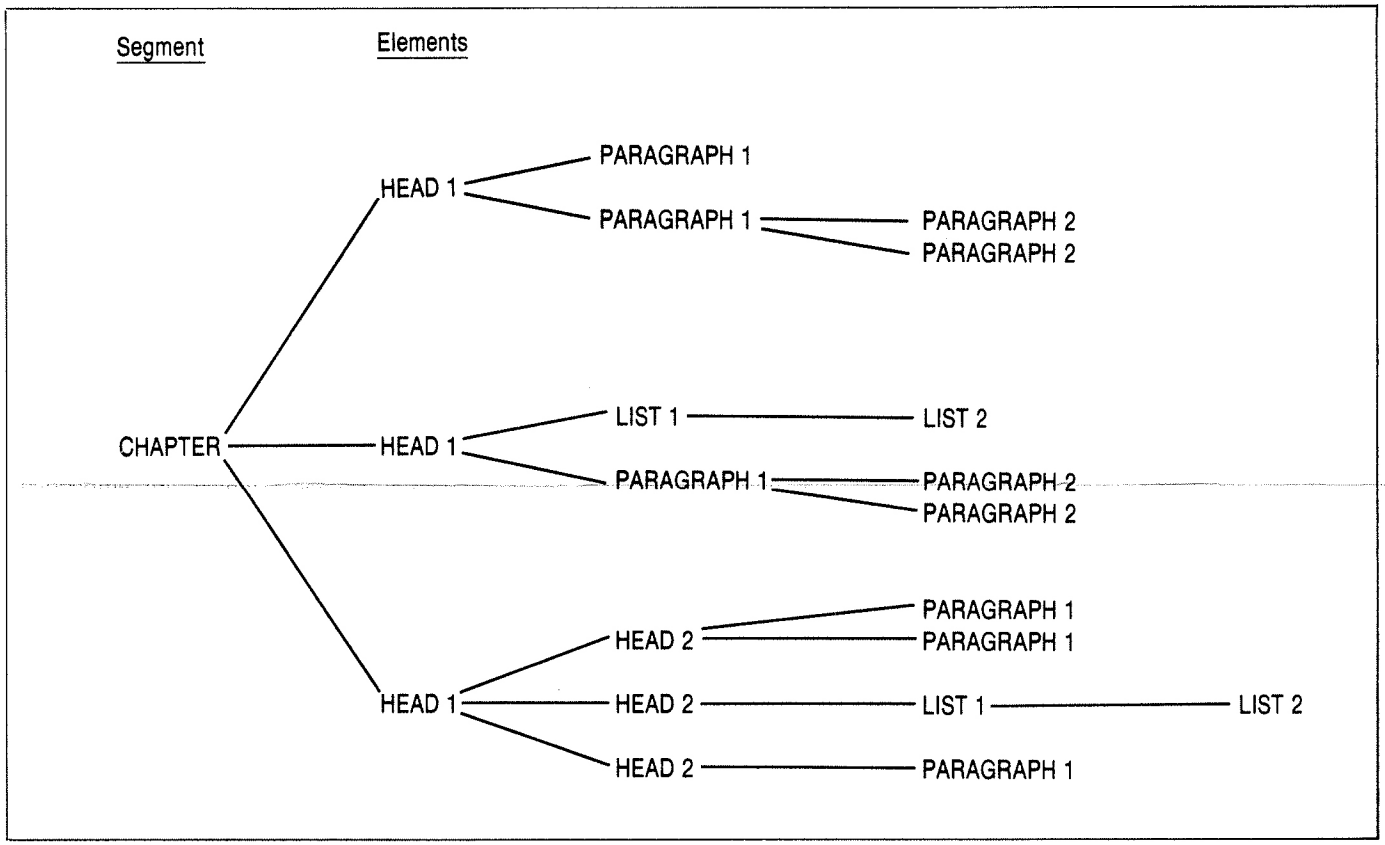

Illustration 2. A sample hierarchy within a chapter

communications and computers is cost effective enough to consider bringing it home. Home computing will go beyond a small personal computer. With simple hook-ups to either phone lines, satellite services, or cable TV, the computer will give us the ability to access a wide range of data. Much of this data will be text. Without standards for everyone to follow — from hardware manufacturers to software developers to end users — we will have a much more difficult path to follow to get there.

Standards give everyone a common direction to follow. They reduce the risk of development costs and protect everyone's investment.

The final goal to be achieved by GenCode will be in role-reversal. The text file is not the by-product of some production process. Text is its own product. Books and other by-products are the only methods of its dissemination. Text with both its physical and intellectual contact is the real value item being captured.

FOR FURTHER INFORMATION

If you would like to find out more about GenCode, or to participate in its development, please contact GCCA at the following address:

1730 North Lynn Street/Suite 604
Arlington, Virginia 22209
Telephone (703) 841-8160

Members of the GenCode Committee

Barbara N. Beeton, American Mathematical Society
Sperling Martin, Aspen Systems Corporation
Charles Lightfoot, Autologic Incorporated
W.W. Tunnicliffe, Tunnicliffe Associates, Inc.
Victor G. Stotland, Bureau of Labor Statistics
Merle Coe, Computer Data Systems, Inc.
Jim Leithauser, Computer Data Systems, Inc.
John S. McMillan, Electronic Keyboarding, Inc.
Donald Rollert, U.S. Government Printing Office
Rich McElroy, U.S. Government Printing Office
Dale E. Bowman, Grolier, Inc.
Dr. Charles Goldfarb, IBM Corporation
John W. Oster, Informatics, Inc.
William W. Davis Jr., Internal Revenue Service
Brian Travis, Internal Revenue Service
Douglas Keeling, RBW, Inc.
Bob Stanard, Riverside Press
Carl J. Buehler Jr., Lehigh/Rocappi, Inc.
Joseph V. Gangemi, Lehigh/Rocappi, Inc.
Jim Bradley, U.S. Department of Energy
Richard Flanagan, Varityper Division, AM International
Nancy Shaffron, Xerox Corporation
Charles Roth, Xerox Corporation
Sharon Adler, Boeing Computer Services
Joseph Johnson, Federal Deposit Insurance Corporation
Bernie Lazorchak, Joint Committee on Printing
Norman W. Scharpf, GCA
Aaron H. Bigman, GCA

Graphic Communications Association

The Graphic Communications Computer Association became the Graphic Communications Association (GCA) on May 27, 1981. The purpose of the Graphic Communications Association is to enhance yields within the graphic communications industry by:

1) Improving communications and coordination among various industry segments.

2) Expanding the application of computers and related electronic technologies within the industry.

3) Developing the use of modern management concepts and techniques.

Graphic Communications Computer Association, an Affiliate of Printing Industries of America, Inc.